

*ARC303*

# *architecture & infrastructure*

Transactional

## Programming In A Windows World

James Johnson and Don Box  
Architects, Indigo  
Microsoft Corporation  
{jimjohn,dbox}@microsoft.com

**PDC**<sup>03</sup>

Make the connection

**Microsoft**<sup>®</sup>

# WinFX Developer Preview

Tools

Visual Studio .net  
Visual Basic .net  
Visual C#.net  
Visual J#.net  
Visual C++.net

## Client Application Model

### Avalon

System.Windows

### Windows Forms

System.Windows.Forms

## Web & Service Application Model

### ASP.NET / Indigo

System.Web

## Data Systems Application Model

### Win FS

System.Storage

### Yukon

System.Data.SqlServer

## Mobile PC & Devices Application Model

### Compact Framework

System.Windows.Forms

### Mobile PC Optimized

System.Windows

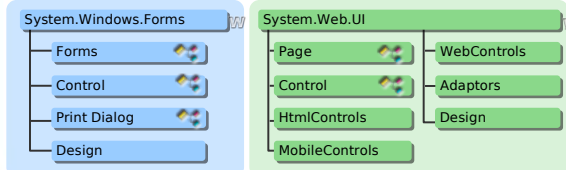
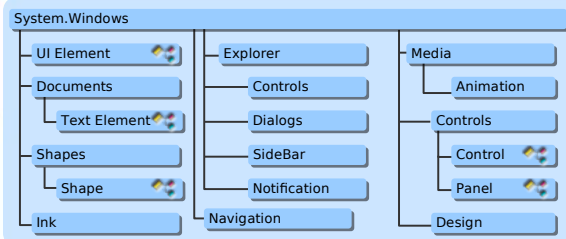
## Command Line

System.Console

### NT Service

System.ServiceProcess

## Presentation



System.Help

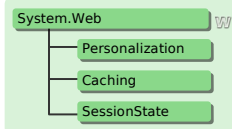
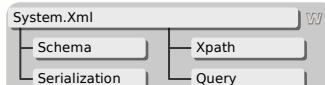
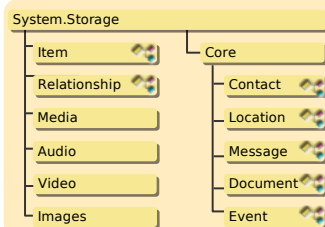
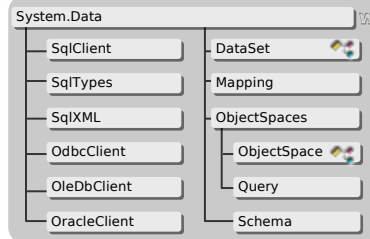
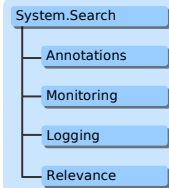
System.Drawing

System.NaturalLanguageServices

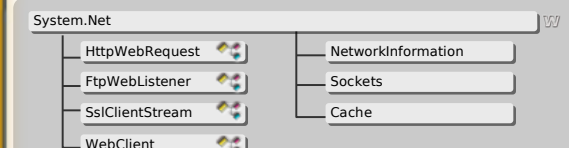
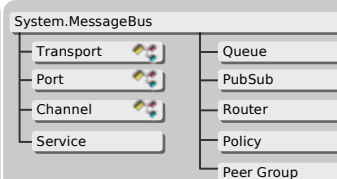
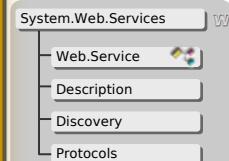
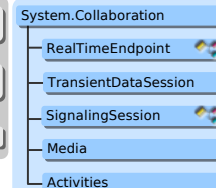
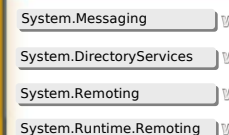
System.Speech

- Recognition
- Synthesis

## Data

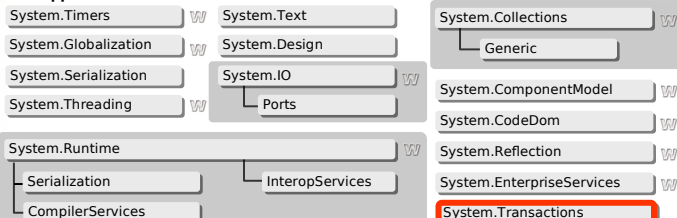


## Communication

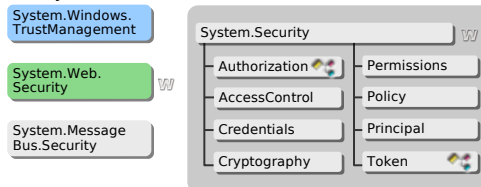


## Fundamentals

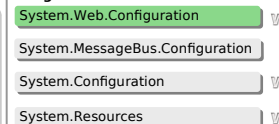
### Base & Application Services



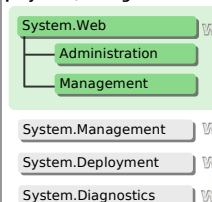
### Security



### Configuration



### Deployment/Management



# Why Transactions?

Order vs. Chaos

**PDC**<sup>03</sup>

Make the connection

# Why Not Transactions?

## Excuses and Issues

# Transactions And Windows

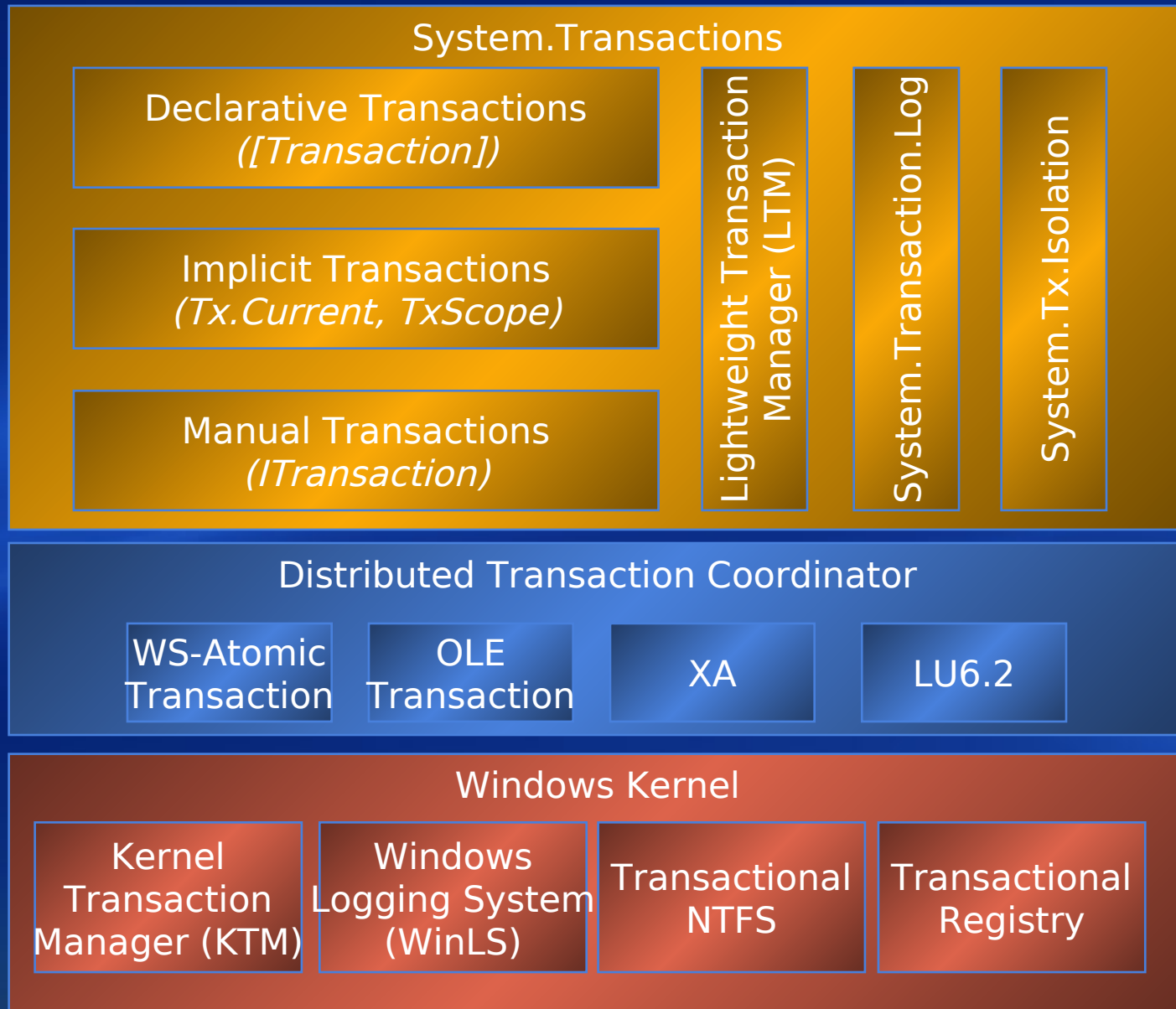
Productivity, Scalability, Ubiquity

**PDC**<sup>03</sup>

Make the connection



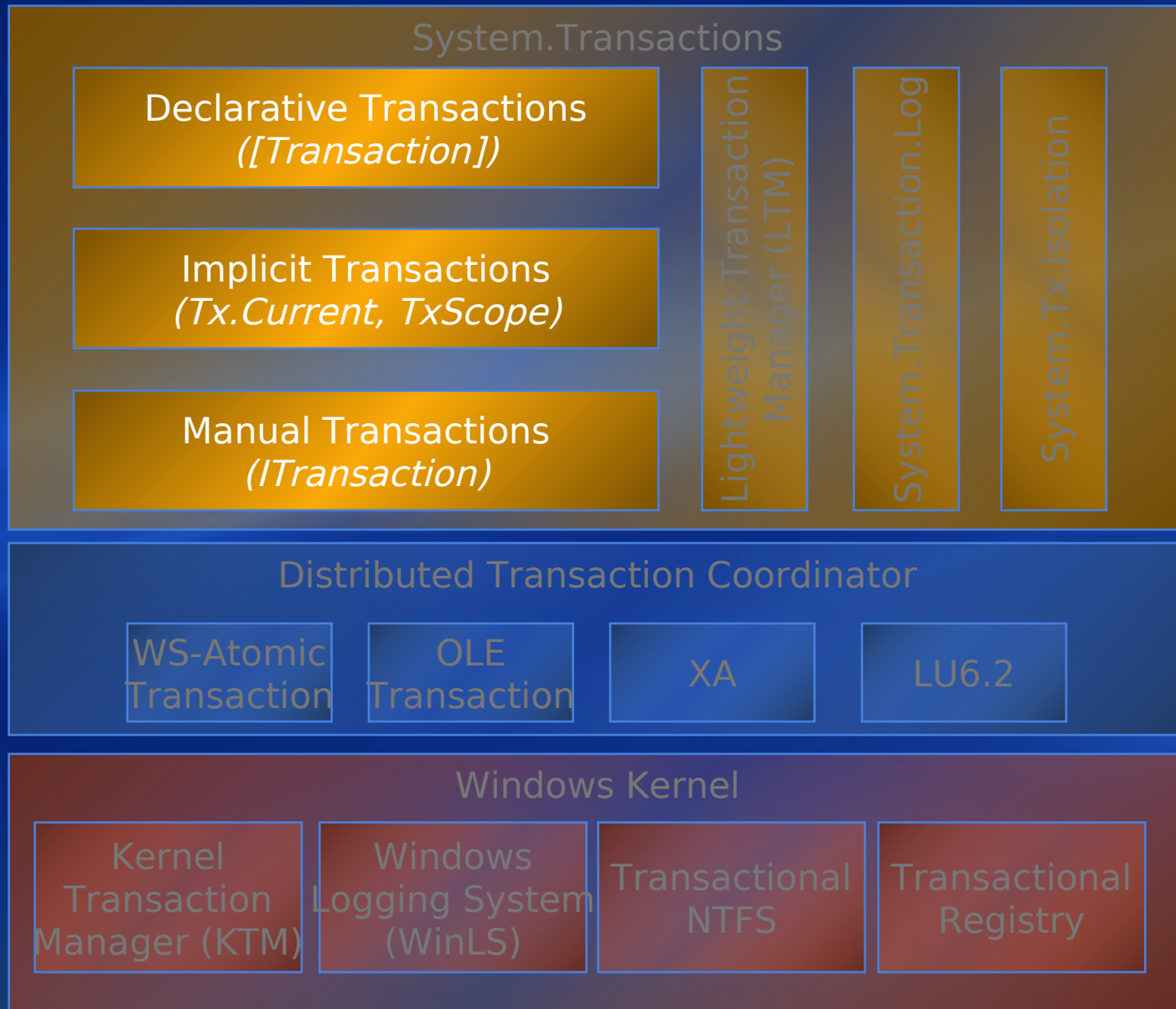
# Transactions And Windows



# System.Transactions

Transactions come to the CLR

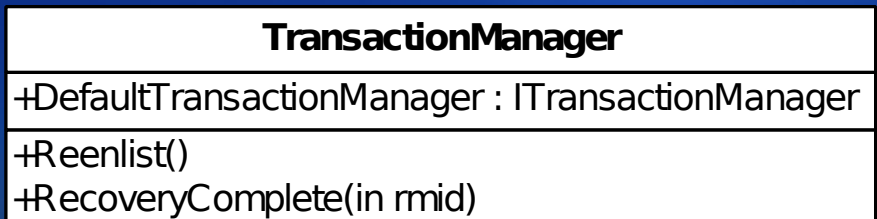
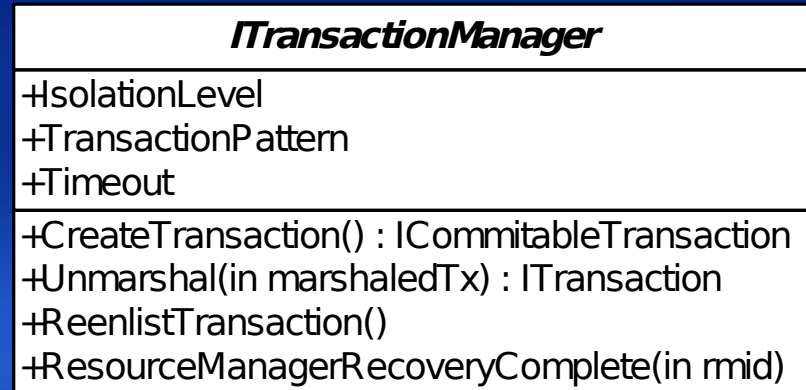
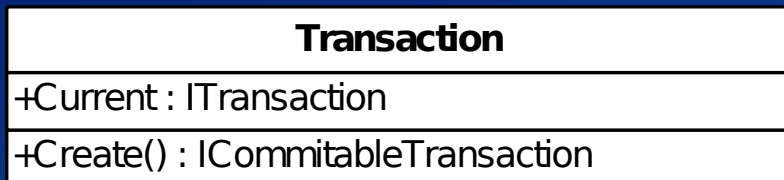
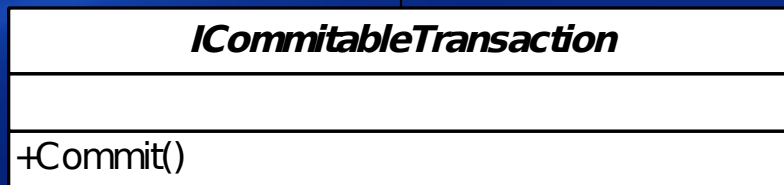
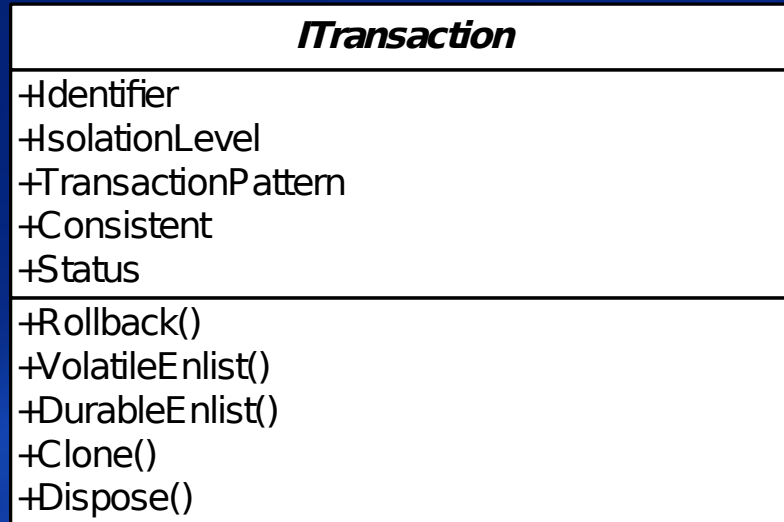
# System.Transactions





# System.Transactions

## Explicit/Manual Model



# System.Transactions

## demo

Kicking the tires...

**PDC**<sup>03</sup>

Make the connection

# System.Transactions Implicit Model

```
public class TransactionScope : IDisposable {  
    public TransactionScope(TransactionScopeOptions opts);  
    public TransactionScope(ITransaction tx);  
  
    public bool Consistent { get; set; }  
    public ITransaction Detach();  
}
```

[Flags]

```
public enum TransactionScopeOptions {
```

```
    TransactionNeeded = 0,
```

```
    NoTransactionNeeded = 1,
```

```
    TransactionOptional = 2,
```

```
    ImportOptional = 0,
```

```
    ImportIgnored = 16,
```

```
    NotSupported = ImportIgnored | NoTransactionNeeded
```

```
    Supported = ImportOptional | TransactionOptional,
```

```
    Required = ImportOptional | TransactionNeeded,
```

```
    RequiresNew = ImportIgnored | TransactionNeeded
```

```
}
```

# System.Transactions Declarative Model

```
public class TransactionScopeAttribute : Attribute {  
    public UsageMode Usage { get; set; }  
}
```

```
public class ContextPropagationAttribute : Attribute  
{  
    public UsageMode Inbound { get; set; }  
    public UsageMode Outbound { get; set; }  
    public Type ContextType { get; set; }  
}
```

# System.Transactions demo

Implicit scopes in Whidbey



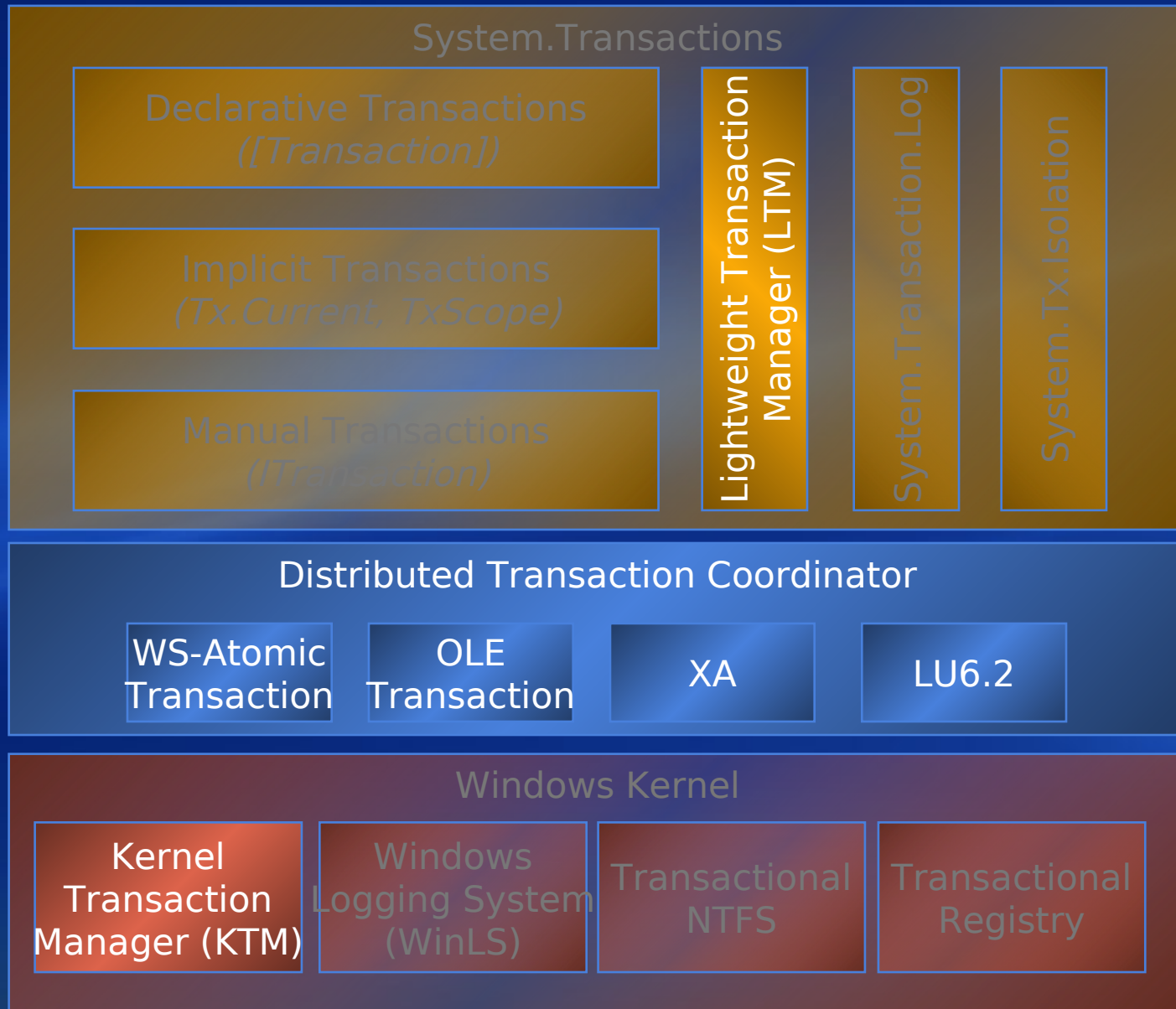
# LTM, KTM And DTC

## The Three Amigos of Transaction Management

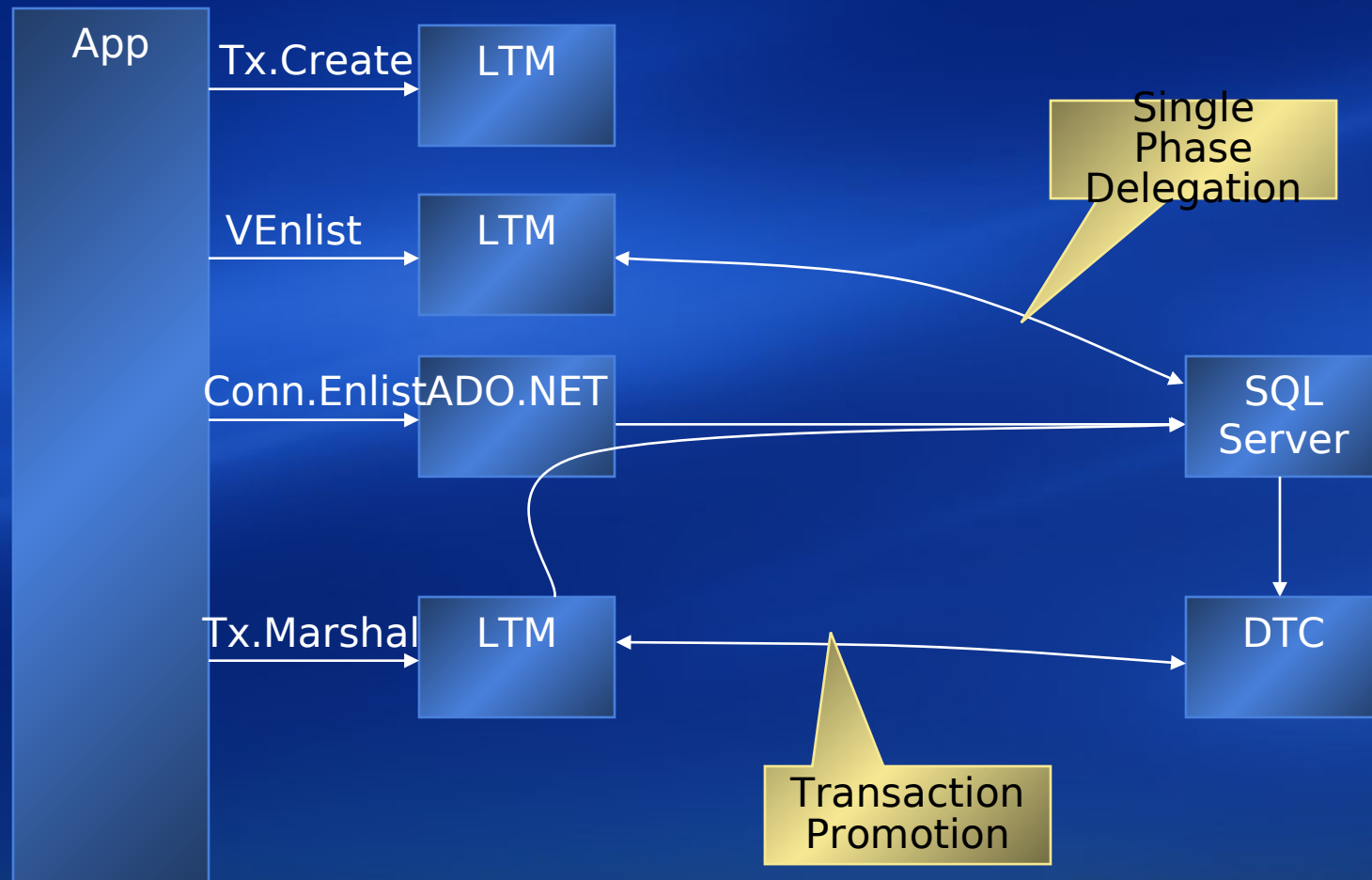


Make the connection

# LTM, KTM, DTC



# Transaction Agility



# Agile Transactions

# demo

Stupid transaction tricks...

# Transactional Resources

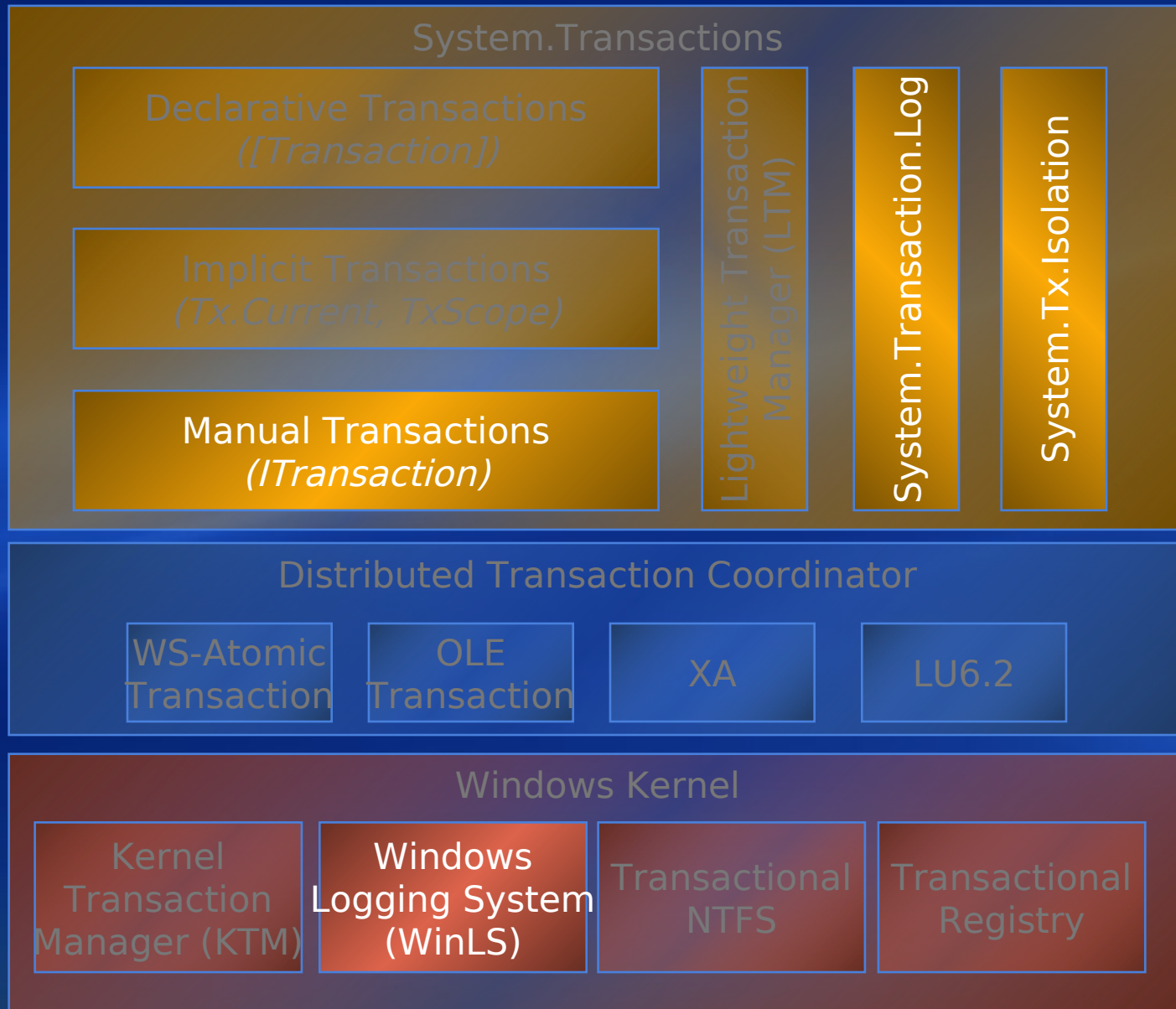
Beyond the DBMS

**PDC**<sup>03</sup>

Make the connection

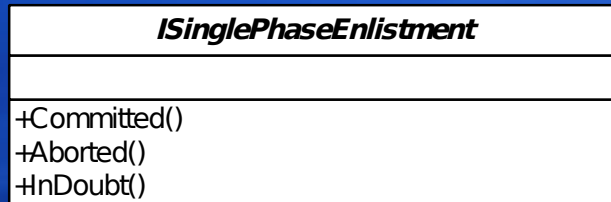
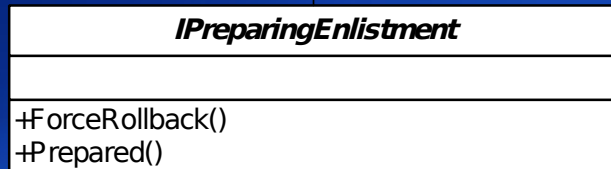
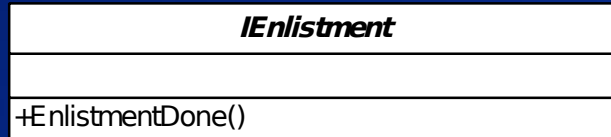


# Transactional Resources

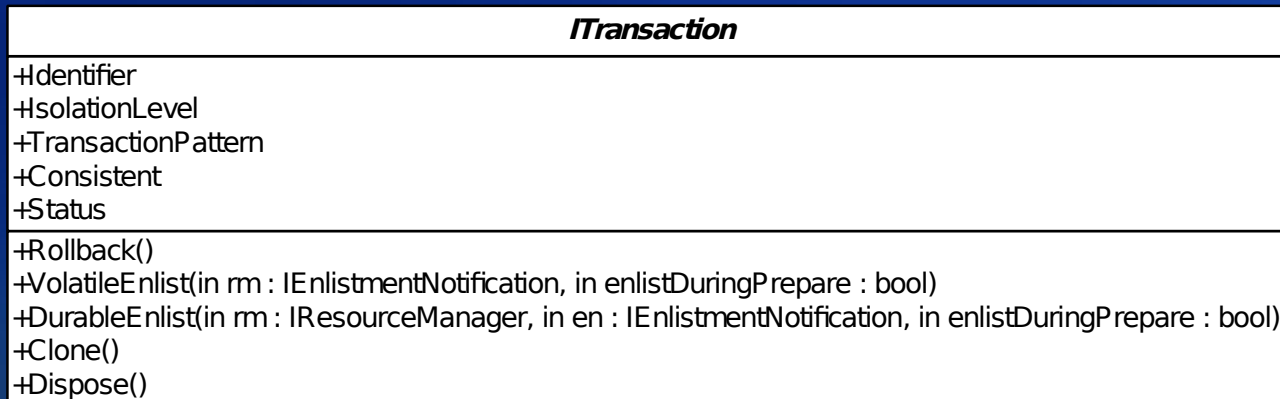
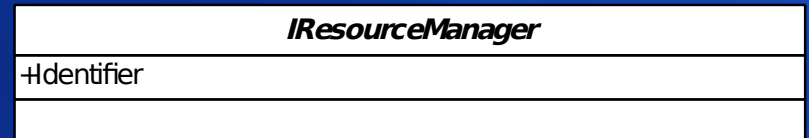
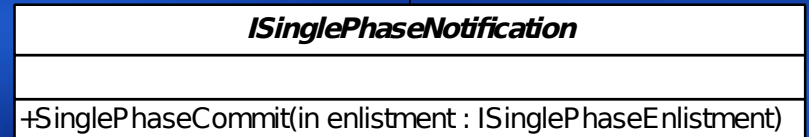
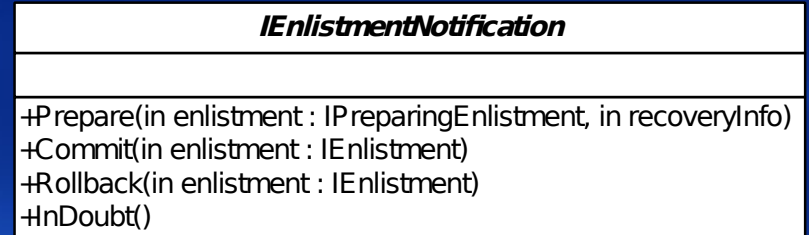


# System.Transaction And Enlistment

## Transaction Manager



## Resource Manager



# Volatile Enlistment demo

Suffrage for VB...

# Transactions And Durability

- Durable resources typically rely on a log
- System.Transactions.Log provides a unified transaction-aware log
  - Uses Windows Logging System (WinLS)
  - Implements ARIES protocol for undo/compensate
  - Savepoints allow checkpointing work
- Common log reduces I/O costs for all participants

# Transactions And Isolation

- Transaction resources typically use locks to enforce isolation
- System.Transaction.Isolation provides an app-level lock for internal use by RM
  - Implements read/update/write semantics
  - Implements two-phase locking protocol
  - Allows arbitrary correlation for reentrancy



# Where Are We?

- System.Transactions makes transactions broadly accessible in Whidbey
- The LTM and KTM improve transaction scalability based on the needs of the application, not the infrastructure
- We're making it much easier to build transactional resources

# For More Information

- Come see us
  - Immediately after this session
  - Web/Services Lounge: 309 Foyer
- MSDN “Longhorn” DevCenter  
<http://msdn.microsoft.com/longhorn>
- Newsgroup
  - microsoft.public.windows.developer.winfx.fundamentals
- At PDC
  - Hands on labs: On-site or download from CommNet
  - Ask The Experts: Tuesday 7 PM – 9 PM Hall G, H
  - PDC Weblogs: <http://pdcbloggers.net>



# PDC<sup>03</sup>

Make the connection

**Microsoft Professional Developers Conference 2003**

October 26 - 30, 2003, Los Angeles, CA

**Microsoft®**